Generic R&D Proposal submitted on June 16th 2017 for FY18:

eRD20: Developing Simulation and Analysis Tools for the EIC

Proponents

ANL (3) Whitney Armstrong, Sergei Chekanov, David Blyth
BNL (4) Elke-Caroline Aschenauer, **Alexander Kiselev (co-PI)**, Jerome Lauret, Christopher Pinkenburg
Fermilab (1) Stefan Prestel
INFN Trieste (1) Andrea Bressan
Jefferson Lab (4) **Markus Diefenthaler (co-PI)**, David Lawrence, Dmitry Romanov, Maurizio Ungaro
SLAC (2) Makoto Asai, Andrea Dotti
William & Mary (1) Wouter Deconinck

Contact

Markus Diefenthaler, Alexander Kiselev

Abstract

The EIC realization will require significant investment from the Nuclear Physics community in the U.S. and around the world. Like all modern accelerator facilities at the leading edge of technology, the computational demands will be sizeable. To realize the physics program laid out in the White Paper and beyond, the high-luminosity machine needs to be matched by detectors capable of delivering motivating science. The success of detector designs depends on our ability to accurately simulate their response and analyze their physics performance. Therefore, early investment in the development of software tools will have an immense impact on the quality of the future scientific output. With this in mind we have proposed in FY17 to identify and develop the required simulation and analysis tools for an EIC in a software consortium. In this proposal, we review our forward-looking global objectives that we think will help sustain a software community for more than a decade. We then identify the high-priority projects for FY18 and present our budget request. In the last part of our proposal, we report on the status of our FY17 projects.

Overview

Objective

The EIC will revolutionize our understanding of the inner structure of nucleons and nuclei. Developing the physics program for the EIC, and designing the detectors needed to realize it, requires a plethora of software tools and multifaceted analysis efforts. Many of these tools have yet to be developed or need to be expanded and tuned for the physics reach of the EIC. Currently, various groups use disparate sets of software tools to achieve the same or similar analysis tasks such as Monte Carlo event generation, detector simulations, track reconstruction, event visualization, and data storage to name a few examples. With a long-range goal of the successful execution of the EIC scientific program in mind, it is clear that early investment in the development of well-defined interfaces for communicating, sharing, and collaborating, will facilitate a timely completion of not just the planning and design of an EIC but ultimate delivery of the physics capable with an EIC.

In our consortium, we aim to develop analysis tools and techniques for the EIC, and facilitate communication and collaboration among current and future developers and users. We will help coordinate the EIC software effort, providing organization and guidance to help seed growth of a software community that will exist for well over a decade. While our localized efforts are typically focused on completing specific tasks or developing certain tools, the consortium will focus also on achieving the following forward-looking goals:

1. **Organizational efforts with an emphasis on communication:** We help with the organization of the software effort for the EIC by providing documentation about the available EIC software and by maintaining a software repository. This function will be eventually taken over by an official EIC software group. We encourage participation in our consortium and have monthly meetings to foster collaboration. Since FY16, we have organized workshops related to EIC software development.

2. **Planning for the future with forward compatibility:** We will continue our involvement in the "Future Trends in Nuclear Physics Computing" workshops to discuss new developments and trends in scientific computing and to identify common goals and a common vision for the EIC software. Incorporating new standards and validating our tools on new computing infrastructures are among the main goals of our consortium.

3. **Interfaces and integration:** Given the current stage of the EIC project, it is too early to define the analysis tools of the EIC. However, it is important to connect the existing frameworks / toolkits and to identify the key pieces for a future EIC toolkit. We will work on interfaces between the existing frameworks / toolkits and aim to collaborate with other R&D consortia and projects in general to integrate their tools into existing frameworks / toolkits. By doing so, we will start to define the key pieces of the EIC toolkit and identify the high-priority R&D projects.

Plan for FY18

In FY18, we are placing emphasis on:

- Monte Carlo Development
- Development of Detector Simulations with a focus on
  - "Geant4 Simulations"
  - "Interfaces and integration"
- Initiatives for:
  - "High-Performance Computing"
  - "Machine Learning"
  - "Web-Based User Interfaces"

We will continue our involvement in the "Future Trends in Nuclear Physics Computing" workshop and will continue building connections to the HEP Computing community via the HEP Software Foundation and the HEP Center for Computing Excellence.

Monte Carlo Development

In the "EIC Software Meeting" in FY16, we have reviewed the Monte Carlo Event Generators (MCEGs) that are available for EIC studies and identified MCEGs and MC libraries that need to be developed:

- For the development of the EIC analysis, a MCEG for TMDs is urgently required. Markus Diefenthaler and Stefan Prestel work on a TMD MCEG based on the Pythia8 MCEG and the DIRE parton shower. Their project is funded by a LDRD project at Jefferson Lab and not included in our proposal.
- There is also work required on MCEG for eA processes. The development of BeAGLE, a MCEG for eA collisions in the nuclear shadowing / saturation regime, is funded by the Generic R&D program for the EIC (eRD17) and also a LDRD project at Jefferson Lab. Elke-Caroline Aschenauer is part of this project.
- Precision measurements require a good understanding of radiative

corrections. Elke-Caroline Aschenauer and Andrea Bressan will continue their development of a library for simulating radiative effects for both polarized and unpolarized observables. The status of their "**Consistent approach to integrate radiative corrections into MCEGs**" and the deliverables for FY18 are listed on page 9.

In FY18, Markus Diefenthaler will initiate a project with the Monte Carlo communities in the US and Europe (MCnet) to review and evaluate the MCEGs available for an EIC. The work will be based on the unpublished results from the "EIC Software Meeting" in September 2015 and the available documentation at BNL and Jefferson Lab and result in a "**Roadmap for an EIC MCEG initiative**". The initiative should encourage a strong interplay between experiment and theory and also connect the MCEG efforts in NP and HEP. The eRD20 funds will be used to initiate a project that can be funded by various sources and will result in EIC related publications.

High-Performance Computing

We would like to take full advantage of advances in scientific computing, in particular the advance of Exascale Computing and start to collaborate with the three ASCR-operated computational science user facilities: the Argonne Leadership Computing Facility at ANL, the National Energy Research Scientific Computing Center (NERSC) at LBNL, and the Oak Ridge Leadership Computing Facility at ORNL. In FY18, we will prepare EIC-related HPC projects and will begin a dialogue with the three ASCR-operated computational science user facilities.

Development of Detector Simulations

Geant4 Simulations Makoto Asai and Andrea Dotti are serving as the point of contact between the EIC Software Consortium and the Geant4 International Collaboration, representing the EIC community needs in the Geant4 collaboration, monitoring the progress, and making sure feedbacks are delivered to ESC in timely manner. One of the most important aspects for the success of the Geant4 simulation of EIC detectors is the correctness of the physics simulation modeling. While the efforts that have been done for the simulation of LHC detectors are a good starting point for the EIC simulation, it is however expected that tuning of the physics modeling will be needed to fully address the energy range of the EIC physics and peculiarities of EIC simulations. In FY18, we plan to work on the following two topics:

- ❏ Define a physics list tuned for the EIC needs. The transition regions between models and the choice of the models to be used requires validation against published data and against alternative MC codes.
- ❏ Identify possible areas of improvement of the simulation of optical photons in Geant4 for its use in RICH detectors.

The FY17 status of our "Validation of critical Geant4 physics in the energy regime of the EIC" can be found on page 10.

Interfaces and Integration

In FY18, we have decided to focus on three projects:

- The progress on our "**Work towards a unified track reconstruction**" is summarized on page 10. Based on our feasibility study, we have decided to develop tracking libraries based on the tracking software developed at ANL, BNL, and Jefferson Lab. In FY18, we will define the tracking libraries and their interfaces and start their development in a sandbox environment compatible with the existing ANL, BNL, and Jefferson Lab frameworks / toolkits. Our modular tracking software will be developed as open source software so it can be easily contributed to by other NP projects. Several of the proponents of this proposal will facilitate adoption and real-world testing of the tracking software at upcoming experiments at BNL and Jefferson Lab.

- The tracking libraries require "**Work towards a common geometry and detector interface**" and a common event model. In FY17, we have made an open document available listing the requirements for a geometry and detector interface for an EIC and the available technologies. Our considerations can be applied to any geometry and detector interface independently of the specific technology choice. However, the focus is on the I/O of geometry and on the link to sensitivity information. These two aspects have been the focus of our discussions. In FY18, we will make a prototype code available for three NP detectors (tracker, TPC and calorimetric hits) and demonstrate that the hits handling in our schema is technology independent by creating a writer for Geant4, a reader for ROOT, and a reader for SciPy/NumPy. We will also demonstrate that the identifiers created in this way satisfy the needs of reconstruction and can be maintained with minimal effort. For the time being, we have decided on GDML as the geometry-exchange format. We will demonstrate that we can export to GDML the information about the identifiers of SensitiveDetectors using current GDML schema and that we can easily port one of the existing frameworks to the result of this project.

- "**Developing interfaces to forward compatible, self-descriptive file formats**" Using ROOT files for MC events exchange does not seem to be a good option because this basically requires ROOT libraries availability on every computer where one would like to read the files in. One can argue that this limitation is "fake", but in reality ROOT is not available on some HPC installations. Native ASCII format, which MC generators use to produce output files, is clearly outdated. As a first step of our R&D work in this direction we investigated feasibility of using existing ProMC file format developed by Sergei Chekanov in conjunction with the HepSim repository. It appeared to us that this format, primarily oriented to store PYTHIA events, was not really well adapted to handle at the same time event files produced by various MC generators with different event-level specific variables. We developed a different format (EicMC), also based on Google protocol buffer message-based streaming logic, but with several improvements compared to ProMC (see FY17 implementation section). Both ProMC and EicMC have similar functionality and their own pros and cons, with ProMC being already adopted by part of the HEP community and perhaps more efficient in terms of storage (at a cost of floating point precision) and EicMC being more feature-rich, flexible, self-descriptive and less dependent on legacy software. Realizing that the choice of file format to store MC events may determine input interfaces for several years ahead, we considered to postpone the final decision and to try to adapt yet another format, widely used by the scientific community, namely HDF5. Based on the feedback we obtained from the developers we are going to implement DIS MC event I/O handling in this format as well and perform a complete comparative review of the available options, including ROOT files as fallback solution. We've got a preliminary agreement of one of the Stony Brook postdocs, who is willing to join our Consortium on the long-term basis, to perform this task on the time scale of 2-3 months, starting in fall this year. The deliverable here will be the final choice of MC event exchange format and its adaptation to HepSim repository usage (in particular adding specific metadata structures, which are required by HepSim hosting server).

Machine Learning

While the term neural networks (NN) has been around for decades, over the last several years neural networks are getting a new spin. Better understanding of basic principles, emerging new methods and faster hardware (GPUs, FPGAs, TPUs) pushed the evolution of deep neural networks (DNN) so that machine learning has made a quantum leap, bringing speech recognition, better image processing, self driving cars and much more.
Nuclear and high energy physics may benefit from this resurgence too, because most of the new approaches may be transferable to both online and offline physical analysis problems. E.g., convolutional and fully convolutional neural networks (CNN, FCN) are efficient and directly applicable to pattern recognition, PID, clustering and parameter extraction of cell based detectors like calorimeters, RICH, cell vertexes and others. Three-dimensional convolutional networks are efficient for volumetric or spatiotemporal pattern recognition which can be applied to PID, DAQ and track finding and fitting. Multimodal deep learning can be used for data fusion - combining outputs from multiple detectors. DNN may also be used for anomaly

detection, providing additional possibilities for data selection. All of the above examples of Neural Networks integrate well with classical analysis methods and with each other. Thus, DNN can be employed in a leading or supplementary role in a large number of tasks.
Because of high scientific and practical attention and support, by way of, large investments from the private sector, a number of high quality, open source, tools for machine learning have emerged during the last several years. However, most of the tools, resources and documentations are dedicated to non-physical data and applications, e.g., image recognition. So an additional API layer which is specially designed for physical data and analysis would be required. There were several attempts to create universal frameworks for nuclear and high energy physics such as hep_ml or MLtools. Unfortunately, the development of the tools stagnated due to their generic nature, limited manpower and lack of support.
This proposal is about creating an EIC-centric machine learning framework. Instead of trying to be general, the development of the framework will be solving narrow, EIC specific, priority ordered problems, utilizing DNN and other modern machine learning methods and tools. We will work with Keras, a machine learning framework with a well-designed API that uses TensorFlow, CNTK, and Theano backends for defining abstract, general-purpose computation graphs. Keras allows the building of deep learning models by clipping together high-level building blocks and running them on backends, which in turn utilize CPUs, GPUs, or even new specially designed Tensor Processing Units (TPUs) in the most efficient way.
While the DNN training phase may be resource intensive, the usage of trained DNN usually is lean and efficient. Moreover, it is suggested that in the near future the backends will have an ability to download trained networks directly to FPGAs, which, as studies conclude may be much more efficient for running neural networks than CPUs and GPUs. Such workflow also looks very promising in terms of DAQ, triggers and real time data processing.

In FY18, we will document selected examples for using DNN in NP and work on a roadmap for an EIC-centric machine learning framework.

Web-Based User Interfaces

Exploring user-centered designs With the growing complexity and size of experiments, the complexity of the analysis environment and also time spent by physicists in dealing with the analysis infrastructure rather than on doing physics grew. Currently, the anecdotal data is that a typical LHC student or postdoc spends up to 50% of his or her time in dealing with computing issues. To improve this situation for the EIC, we have started to discuss new ideas for analysis environments. One key idea in beginning to think about future analysis environments is to understand the user requirements of the analysis environment first and foremost. This requires the engagement of the wider community of physicists whose primary (and perhaps not even secondary) interest is not in computing. While we are still working on a strategy for a meaningful survey of user requirements, we have started to explore alternative ideas for user interfaces by working on web interfaces. In FY18, we will provide a web interface for the basic functionality of one of the simulation tools for the EIC. The ongoing summer project by a University of Connecticut student will be the starting point for this project.

Universal web-based event display We aim for an universal web-based event display motivated by three use cases:

- **simulations and reconstruction experts** fast reconstructed event visualization for early problem detection.
- **scientists performing data analysis** use reconstructed event visualization, detector geometry internals visualization, and require high-quality raster or scalable vector illustrations for papers and posters.
- **public outreach** needs are represented by STEM-teaching professors, media and news agencies, who would like to get a quick insight into the live experiment's life.

At the software consortium meeting in May 2017, plans for future development of an Event Display (EVD) were presented by the STAR group, following a progress report on the current implementation and adaptation of the STAR WebGL EVD (see section "**Progress Report"** for more information). Provided funding, we will be able to realize our ambitious goal of an unified event display with the STAR group. A work plan would comprise:

- Determine the complete properties of a graphical user interface, ensuring maximal coverage of existing devices, platforms and operating systems
- Explore user interaction patterns best suited in the context of the universal event display and large international collaboration of scientists and engineers
- Complete the implementation of the primary geometry input format (see section on geometry and detector interface)
- Research and develop the event description and event meta-data format
- Finalize the design and implementation of the responsive, platform-aware layout of the graphical interface
- Provide package documentation including extensive set of user and developer examples
- Perform code review of the final product to ensure proper modularity of the components

Funding request for FY18

A focused effort is essential for any productive R&D work. We request a travel budget of **USD 70,000** to allow proponents to meet and to work together on key tasks or to invite visiting scientists that are essential to the R&D effort. Part of the travel money will be used to attend important conferences and workshops related to our work, in particular to support the proponent's travel to the "Future Trends in Nuclear Physics Computing" workshop in 2018.

We request **USD 30,000** to fund undergraduate projects. The undergraduate students will work mainly in the summer on key tasks of our project.

Nominal budget In total, we request a FY18 budget of **USD 100,000**.

Nominal budget minus 20% We would compensate a budget reduction by USD 20,000 by only three instead of four in-person meetings and a strict limitation of travel to conferences and workshops. This would limit the outreach of the EIC Software Consortium.

Nominal budget minus 40% We would compensate a reduction of our budget by USD 40,000 by only two instead of four in-person meetings in FY18, no support for travel to conferences and workshops, as well as we will be very limited in resources to attract skilful manpower with strong coding abilities, as well as young computing-oriented scientists to join our efforts. As a matter of fact this would allow for no outreach of the EIC Software Consortium and would inevitably cause re-adjustment of the ambitious goals we are going to put in front of us for the next year period.

Progress Report

The work of the EIC Software Consortium has started in FY17. In our proposal, we have presented seven projects.

- ❑ Consistent approach to integrate radiative corrections into MCEGs
- ❑ Validation and tuning of critical Geant4 physics in the energy regime of the EIC
- ❑ Promote open-data developments for efficient data-MC comparisons
- ❑ Work towards a common geometry and detector interface
- ❑ Work towards a unified track reconstruction
- ❑ Developing interfaces to forward compatible, self-descriptive file formats
- ❑ Start the development of a universal event display for MC events

We have made so far no progress on our work on "**Promote open-data developments for efficient data-MC comparisons**". The status of our "**Work towards a common geometry and detector interface**" is presented in "**Plan for FY18**".

Consistent approach to integrate radiative corrections into MCEGs

We have identified the radiative correction code HERACLES embedded into DJANGOH as the best starting point to develop the standalone radiative correction code library and are currently working on the following points:

- extract HERACLES from DJANGOH
- write a fortran and C++ wrapper to make the library flexible to be integrated into different MC codes
- (and most importantly) establish a testing procedure to verify the MC-radiative correction approach gives the same results as dedicated standalone process dependent MC calculations. To execute this test for several different processes, i.e. inclusive DIS, semi-inclusive DIS and exclusive reactions is our next milestone.

We are in contact with Dr. H. Spiessberger, the author of HERACLES, aiming to establish a collaboration for this project.

Validation of critical Geant4 physics in the energy regime of the EIC

The technical aspects of the applications needed to perform physics validations have been completed. A *single interaction application* has been created and regression-testing macros prepared for relevant physics processes. We have made the application publicly available on our repository. An existing second application tailored to CPU-performance measurements has been generalized beyond the initial HEP domain. The application can read GDML files and thus could be used to test EIC geometries. We have created a Geant4 standalone example for reading EicMC and ProMC data files. The example code gives an idea of our strategy on how to read the data files in a multithreaded application.

Work towards an unified track reconstruction

In FY17 we were planning to perform a detailed feasibility study of extracting existing reconstruction codes (either implemented in EIC-related frameworks already or available as third-party packages) into a standalone library, which - provided proper interfacing - could have then been used as a core tracking engine in all software frameworks, which a growing EIC community has in use to date.

It was clearly stated in the proposal that the ultimate success of this project was critically dependent on the progress we make in terms of the definition of a common geometry interface, in particular a subtle issue of coupling simulated detector hits to physical volume instances in a given geometry interface paradigm, as well as on the progress we make with defining other elements of persistency model, in particular definition of hit classes to enable information exchange between simulation, digitization and reconstruction stages of our typical Monte-Carlo workflows.

It turned out that deciding on geometry interface model is a more difficult task than we anticipated originally, and despite obvious progress we made in this direction it is too early to start any subsequent software development (like unified tracking library), which per definition strongly depends on the final decision(s) about geometry handling. Hit structure definitions would perhaps require less work to be implemented, but given the very limited manpower in the consortium and lack of allocated resources to hire people to do the actual coding we decided to focus on identified problems (and solve them) one by one rather than to scatter our resources without real need.

Therefore to this moment only a relatively independent on the geometry fraction of tracking codes is pulled out of the EicRoot source codes and uploaded onto our https://gitlab.com/ESC GitLab repository as a "seed" of the future standalone tracking software library. These codes represent themselves as a recursive Hough-transform-based tree search algorithm for track finding, and were recently used to simulate tracking environment of the STAR forward upgrade within EicRoot framework.

Depending on the progress with geometry and hit structure definitions we may still want to perform the feasibility study, which was mentioned at the beginning of this section. By that time the actual implementation of the unified tracking codes (our second possible deliverable for FY17) does not make sense and we rather consider to carry this task over to FY18.

Developing interfaces to forward compatible, self-descriptive file formats

One of the key deliverables of the first year of the consortium work was setting up of a HepSim repository for an EIC and evaluating our options for the Monte-Carlo event generator file format(s).

We have set up a HepSim server at Jefferson Lab. The main activity related to the HepSim public repository included several improvements in the interface of this database as well as changes that were required to make this database friendly to the EIC project:

- HepSim web interface now includes the new menu "Electron-Ion Collider" which retrieves generated Monte Carlo file samples with different center-of-mass energies proposed for the EIC project
- There is a separation of "reconstruction tag" and "detector tags", such that multiple reconstruction tags can correspond to one detector
- The reconstruction tags can be listed using the main menu
- Detectors can be listed using a dedicated menu
- There is an additional page that collects all validation scripts for ProMC and LCIO files

- ❏ Each detector geometry supported by HepSim includes dedicated web pages. Such pages include images of detectors, and several file formats with detector geometry (such as GDML, ROOT, LCDD, Pandora) for download
- ❏ In collaboration with jsROOT developers, an interactive 3D viewer of detectors have been implemented. It includes a 3D viewer, and an interactive GManager that allows to look at separate detector volumes

In addition to the improvements to the HepSim interface, the ProMC file format was modified to allow more flexibility in storing arbitrary information on events. Each event can store two arrays (one for integer numbers, and one for double-precision numbers) which can be used to keep information on generated events (such as Q2, x_Bj etc). ProMC also includes an "EIC" type of library which is better configured for EIC events, compared to the standard "generic" "promclib" library. In addition to the static libraries, shared libraries can be compiled during the installation step.

We also explored other file formats for Monte-Carlo event exchange. In particular we developed from scratch a new google-protobuf-based file format (EicMC), which is also publicly available from our repository https://gitlab.com/ESC/EicMC.
This file format has a number of distinguishing features. In particular it does not depend on any external legacy libraries except for Google protocol buffer ones, it provides more efficient streaming compression on event collections rather than dealing with individual events, has DIS-generator-specific event-per-event fields implemented for all generators we used so far (and the shared part of the event records is the same for all generators). It provides forward and backward compatibility, as well as true self-descriptiveness (file format is automatically encoded as part of the binary file header). There are interfaces to HepMC2 and Pythia8. EicMC library provides efficient direct access to the individual event records in additional to the sequential streaming. If external dependency on the light-weighted EicMC I/O library is undesirable, one can import .eicmc binary files from his/her codes without a need for this external dependency, just using few dozens of lines of basic C++ codes in a cut'n'paste fashion. In particular it was a very easy exercise to import these binary files from within the Geant4 framework.

We also considered HDF5 scientific file format, which is widely used far beyond the field of physics. At present this is our possible fallback solution in addition to our in-house ProMC and EicMC format developments. HDF5 advantage is the large user base, strong developer team, self-descriptiveness. And overall it is a very light-weighted library with no external dependencies. The format is tuned to run efficiently on the HPC platforms. We were in touch with one of the core developers and got his insight and advice on a possible use of this format at an EIC. We identified a Stony Brook postdoc who agreed to implement EIC MC event generator exchange format based on HDF5 and perform comparative study between all formats we have at hand: ASCII, ROOT, ProMC, EicMC and HDF5. We are going to perform this study by the end of this year and then make our final decision.

Start the development of an universal event display for MC events

One area of our activity was planned to be focused on creating a generic event display for viewing generated (and detector reconstructed) events from any web browser. Work targeted for evaluation comprised the CMS approach and Paraview based on WebGL. While the available funding support did not allow for a thorough prioritized and dedicated time for this evaluation, the joining of the RHIC/STAR collaboration to the ESC provided an opportunity to implement a proof of principle of the Web based approach. STAR had implemented (since 2010) an event display based on Three.js and WebGL with the same principle of no software installation needed on the client side.

The work was discussed internally and further presented at the software consortium meeting (May 2017) where a prototype implementation for the EIC was showed and demonstrated as an adaptation of the STAR EVD (available at http://online.star.bnl.gov/aggregator/livedisplay/. The STAR version is able to show collision event in Real-Time due to its already available and implemented "event reco service" back end and MemCache architecture). Loosely (and historically) based on a full set of Geant3 geometry shapes, the presented prototype showed to already have features such as zooming, standard rotation, 3D views via Google Cardboard and VR sets support and controls to enable/disable the display of volumes or tracks and events. The Events and Geometry were converted from the original EIC geometry using a provided custom JSON-based format, implemented for fast processing (see http://www.eicug.org/display/). An sPHENIX geometry based EVD was also demonstrated and showed (available at https://www.sphenix.bnl.gov/display/). The initial prototype was found to be adequate but relying on aging technologies and hence, is in need for further strengthening. Particularly, the following technology areas have been identified as being in need for a revisit:

- Use of Geant3 would benefit from GDML / AGML import and export
- To support future detector design, support for a wide(r) set of Geometrical Shape types is needed
- Three.js replacement by newest JavaScript standard is needed (ES6+)
- SaaS paradigm is needed and best approach needs to be evaluated
- We would benefit from increased interactivity such as enhanced service information, volume manipulation (STAR has extensive experience in requirements in this area)

We described the path to modernization in our FY18 development plans.